

APPLIED AND GENERATIVE AI

Foundations of Prompt Engineering

Level: Foundation • 2 days (expandable to 3) • Virtual, In-person

Overview

The single biggest factor in what you get from a large language model is what you ask it, yet most people prompt the way they search: a few words, take what comes back, move on. The hard part of prompting is not learning tricks, it is learning why the model responds the way it does, so you can write requests that reliably produce what you actually need, and fix them when they do not.

This is a hands-on, foundation course. It builds prompting skill in the order it actually develops: first a plain mental model of how a model responds to a prompt, then the anatomy of a well-built prompt, then the small set of core techniques that account for most real-world results. From there it turns to what separates casual users from skilled ones: diagnosing and fixing prompts that fail, applying proven patterns to everyday tasks, and knowing when to trust the output. The course teaches fewer techniques than a survey would and practices each until it is a habit. Every module includes a lab, and each module builds on the one before it.

Who Should Attend

- Professionals in any role who use ChatGPT, Claude, or Copilot and want consistently better results
 - Analysts, writers, and knowledge workers making AI a real part of their workflow
 - Anyone whose team expects them to be effective with generative AI tools
- Developers who want to use prompting inside applications and APIs should take *Prompt Engineering for Developers*.

Prerequisites

- No technical background is required
- Some exposure to a tool like ChatGPT or Claude is helpful but not assumed
- Learners who want a grounding in what generative AI is should consider *Introduction to Generative AI* first

What You Will Learn

- Explain, in plain language, how an LLM responds to a prompt and why wording changes results
- Build prompts with a clear structure: role, instructions, context, examples, and output format
- Apply core techniques including examples, step-by-step reasoning, and format constraints
- Diagnose a prompt that is not working and revise it systematically
- Use proven prompt patterns for summarizing, extracting, transforming, and drafting
- Judge when to trust model output and how to check it

Course Outline

Day one: how prompting actually works

- How a Model Responds to a Prompt

- A plain-language mental model: prediction, context, and instructions
- Why small wording changes produce large differences
- The context window: what the model knows about your conversation, and what it forgets
- Lab: run controlled variations of one prompt and observe exactly what changed the output
- Anatomy of a Good Prompt
 - The building blocks: role, task, context, constraints, and output format
 - Being specific: replacing vague asks with measurable ones
 - Giving the model the information it cannot guess
 - Lab: rebuild a set of weak prompts using the full structure and compare results
- Core Techniques
 - Showing, not just telling: examples that teach the model the pattern you want
 - Asking for step-by-step reasoning on hard problems
 - Constraining format: lists, tables, length, and tone
 - Lab: apply each technique to the same task and identify when each one earns its keep

Day two: reliability and real work

- Diagnosing and Fixing Prompts
 - Reading a bad output for clues about what went wrong
 - Systematic revision: change one thing, test, repeat
 - Splitting one big ask into a sequence of smaller ones
 - Lab: take three failing prompts and repair each with a documented diagnosis
- Prompt Patterns for Everyday Tasks
 - Summarizing and extracting information from documents
 - Transforming: rewriting, reformatting, and changing tone
 - Drafting and critiquing: using the model as a first-pass writer and a reviewer
 - Lab: build a small personal library of prompts for your own recurring tasks
- Trust, Verification, and Responsible Use
 - Hallucination: why models state falsehoods confidently, and how to check
 - What not to paste into a prompt: sensitive data and workplace policy
 - Knowing when a task should not be delegated to a model at all
 - Lab: fact-check a set of convincing model outputs and catch the errors

Extended Version

The three-day version keeps the same gradient and adds depth and applied practice:

- Longer, multi-step prompt workflows that chain several prompts together
- Prompting across tools: ChatGPT, Claude, and Copilot differences that matter
- Deeper practice with role-specific scenarios drawn from the class
- A capstone in which each learner builds and tests a complete prompt toolkit for their own role